Machine Learning Modelling

Early Incident Identification

Author

Contents

1 Introduction 3

1.1 Context and Motivation 3

1.2 Problem Statement 3

2 Literature Review 4

3 Methodology 4

3.1 Dataset Description 5

3.2 Data Cleaning Steps 5

3.3 Data Balancing 5

3.4 Classifiers 5

3.4.1 Random Forest 5

3.4.2 Logistic Elastic-Net Regression 6

3.5 Hyper-parameter tuning and Cross validation 6

3.6 Evaluation Metrics 6

4 Results 6

4.0.1 Accuracy 6

4.0.2 False Positive Rate 7

4.0.3 False Negative Rate 7

5 Discussion 7

6 Conclusions 7

7 References 8

8 Appendix 9

8.1 Tables 9

8.2 Confusion Matrix Percentages 10

8.2.1 Logistice Elastic Net Balanced 10

8.2.2 Logistice Elastic Net Unbalanced 11

8.2.3 Random Forest Balanced 11
8.2.4 Random Forest Unbalanced 12

1 Introduction

Computer systems and networks are often under attack from intruders, of whom masquerade under different duress to avoid detection. Their modes of operation vary over time, with advancement in technology within general and individual systems and networks, making it more harder for early detection of their intrusion. There may not be complete assurances that vulnerabilities within systems can be detected early or fixed up on time, given that detection and risk assessment is a continuous process. Moreover, patching up vulnerabilities within computer systems and networks does not imply that all is fixed as modes of operation change based on a number of factors.

According to Lipson (2002, p. 7), communication across the computer systems that are acting as hosts, connected via wired or wireless links, is through the standardized regulations and rules known as Transmission Control Protocol/Internet Protocol, TCP/IP, with IP packet contents being data.

1.1 Context and Motivation

Attacks and intrusion often times may lead to unauthorized access, that may extend unnoticed over time, infection, modification and even the shut-down of the systems and networks. Changes in the systems often times go un-noticed until complaints are raised up on the efficiency of the systems and networks as argued by Lai & Hsia (2007). Successful transmission of the IP packets to the destination and without any interception, within a specified time is highly desired. However, as more systems join up to use a network, the complete security of systems becomes complex, with risk evaluation on the systems encompassing, but not limited to how to certify intrusions' risky levels, the vulnerability and type of vulnerability, the host systems facilitating the intrusion and its state, the level of security being offered to users of the systems and the level of patching of vulnerabilities and their effectiveness.

1.2 Problem Statement

In a realistic setting, not all solutions to intrusions and vulnerabilities are easy to detect or solve and on time. The flow of information packets on the systems can provide information on intrusion, given that certain events may be deviating from the overall norm. Guiding our analysis is the Disc Consulting (DCE) data on IP Packet event records, that has identified malicious and non-malicious event record within their networks and computer systems, for which the attacks are more sophisticated than previously experienced attacks, with no clue on the methods used for intrusion.

A suggested approach is incorporate a real-time threat detection system based on machine learning classification models to categorize an record event as either malicious or not based on IP packets attributes. Performance measures and model selection, will be assessed inline with the scope of the DCE objectives.

2 Literature Review

Lipson (2002, p. 7) outlines the transmission of information from addresses of the source to destination via the IP packets, going through network ports that are designed to specifically handle the information based on the type of information being delivered. Routers facilitate the flow of IP packets through the network, through the best path, in a process known as a hop, with a given number of routers required to handle and move the IP packets closer to the destination. The flow of IP packets via routers terminates if the IP packets reach the source to destination or up until they are forwarded through the pre-specified number of routers as contained in the routing table of the router. The transmission protocol on the other hand ensures the IP packets, in the right order reach their destination, failure to which the re-transmission has to take place. IP addresses on the other hand are unique to each host on the network, facilitating the flow of IP packets from one host to another.

In terms of individual level security Lipson (2002) states that as the number of individual users of computer systems rise, and with no expertise in ensuring their host computers are secured, the level of security offered has fairly been on a decline, as any one without expertise can operate a computer's basic functions, without time to time re-evaluation of the users host security level. Furthermore, they argue that as more demand for high speed internet bandwidth come up with advancement in the internet network, the flow of IP packets is not ingrained in the devices for long, as such critical information that might need re-evaluation gets lost within a short time.

The ability to track down sources of intrusion via IP addresses keep on evolving, with organisations keeping logs on the flow of information in the event that they are required by law enforcement agencies. However, (Hofstede et al., 2017) argue that intruders may opt to delay intrusions intentionally or intrude from different computer systems to avoid being detected. Our analysis is guided by the need to evaluate IP packet data, as such the argument supports the need for real-time threat detection on the available data.

3 Methodology

The data used in this analysis is a network and systems event record data of the Disc Consulting Enterprises (DCE). Each of the data observation is of an individual network packet, recorded by the SIEM logs, after triggering of an event during TCP communications between sources and hosts. The observations are further categorized as malicious or non-malicious, with presence of imbalance biased towards the malicious events, among the categorizing variable. The training of the models on train data takes into consideration the imbalance, as such the influence it has on the model accuracy is assessed. For solving the nature of response variable, balancing of the data is undertaken through up-sampling of the minority class via bootstrapping to have 2 train data sets and one test data that brings out a better representation across for the biased malicious event relative to the non-malicious event. The balanced train data sets are trained extensively using the random forest and logistic elastic net classification models whose parameters are hyper tuned. Cross validation is explored to obtain an optimal model for better learning and predictions (F.Y et al., 2017).

3.1 Dataset Description

The data comprises of 502159 events with 12 predictor variables as in Table 5, and dependent `Class` indicating whether an event trigger is malicious or non-malicious. The independent variables are Assembled Payload Size, DYNRiskA Score, Response Size, Source Ping Time, Operating System, Connection State, Connection Rate, Ingress Router, Server Response Packet Time, Packet Size, Packet TTL, Source IP Concurrent Connection.

3.2 Data Cleaning Steps

`TIPV6 traffic (binary)` was excluded on import as it contains significant proportion of invalid values. The data was cleaned by filtering out to remain with observations whose Class label is either 0 or 1. Categories were merged in certain variables as they represent almost similar umbrella labels, with their individual counts being merely too little to influence the learning of the algorithm. In the case of feature `Operating.Syatem`, the `Windows` category was combined into category `Windows_All`, `iOs, Linux (Unknown) and Other` into category `Others`. The variable `Connection.State` categories were combined with categories `INVALID, NEW and RELATED` forming the `Others` category. Complete cases of observations were selected, as missing data can affect model perfomance. The clean data comprises of 502159 observation.

3.3 Data Balancing

The target variable `Class` indicating whether an event trigger is malicious (3011 observations), `(coded 1)` accounts for 99.4 % of the data, while or non-malicious (499095) observations, `(coded 0)` accounts for 0.6 %. Up-sampling is undertaken on the data by:

1. The separation of observations by class variable labels into different sets
2. Re sampling of the minority class is undertaken, by sampling with replacement a given number of observations per category so that minority class is at a reasonable frequency.
3. Combining of the up-sampled data sets.

In our analysis, the observations in the balanced train data are 39600, the unbalanced train 20000, and the test data 472036. The balanced data has 19800 non-malicious and 19800 malicious observations while unbalanced has 19800 non-malicious and 200 malicious observations.

3.4 Classifiers

Classification models Logistic Elastic-Net Regression and Random Forest‘ map observation features, by using distinctive characteristics, into either of the malicious and non malicious categories.

3.4.1 Random Forest

The random forest algorithm creates a split at each node, based on subset of independent variables randomly selected at individual nodes, as stated by Liaw & Wiener (2002). James et al., (2014, p. 320) defines the steps to building a random forest as:

- The growing of specified number of tree bootstrap from the supplied data
- With each bootstrap sample, a given number of predictors are sampled at each node to select the best split by variables.
- Aggregation of final predictions from the number of trees initially supplied.

3.4.2 Logistic Elastic-Net Regression

The model net extends the ridge and lasso regression by penalizing of `L1 and L2` penalties with an objective of keeping the prediction error rates at a minimum (James et al., 2014, p. 131).

3.5 Hyper-parameter tuning and Cross validation

Hyper-parameters define the structure of the model, and their values cannot be directly derived from from the data used in training according to Yang & Shami, (2020), with the random forest hyper-tuned on the number of trees and the number of variables taken as sampling candidates at each of the node splits and the logistic elastic net hyper-tuned for values of lambda (regularization amount) and alpha (penalty) (Yang & Shami, 2020).

The k-fold cross-validation randomly partitions the train data set into `k` groups of equal size, with `k-1` groups of the set used for training and remaining one used as test set, and the result being mean of the results (A. Ramezan et al., 2019).

3.6 Evaluation Metrics

Metrics comparing model performance, are confusion matrix, the false positives (number of non-malicious events identified incorrectly as malicious), false negatives (number of malicious events identified incorrectly as non-malicious), the accuracy $= \frac{\text{TP+TN}}{\text{TP+FP+TN+FN}}$, precision (Positive Predictive Value, PPV) $= \frac{\text{TP}}{\text{TP+FP}}$, recall (Sensitivity, hit rate, True Positive Rate, TPR) $= \frac{\text{TP}}{\text{TP+FN}}$ and the F1-score $= 2 \times \frac{\text{Precison}\times\text{Recall}}{\text{Precision+Recall}}$. Precision is proportion of malicious correctly identified to sum of malicious correctly identifies and non-malicious incorrectly identifies while recall is maliciously correctly identified to sum of of malicious correctly identified and incorrectly classified malicious and f1 score the harmonic mean of the two.

4 Results

Table 1 provides the optimal parameters, Table 2 the prediction metric summaries, for all models and data sets. Table 3 provides the prediction metrics on the test data for the metrics.

4.0.1 Accuracy

DCE desires higher proportion of correct placing of malicious event records, with the best result being of model, data combination of Elastic Net Unbalanced having an accuracy of 0.9979, model, data combination of Random Forest Balanced having an accuracy of 0.9993. Similarly, in terms of data used, the Balanced, model combination of Random Forest Balanced had the accuracy of 0.9993, while Unbalance data, model combination of Random Forest Unbalanced had the an accuracy of 0.9987.

4.0.2 False Positive Rate

DCE seeks a lower the chance of marking an event malicious when it is non-malicious, with the model, data combination of Elastic Net Balanced having the low FPR of 0.0104 and Random Forest Balanced with lowest FPR of 3×10^{-4}. Based on the data, the data, model combination of Random Forest Balanced having the low FPR of 3×10^{-4} and Elastic Net Unbalanced having the low FPR of 0.

4.0.3 False Negative Rate

The lower the chance of a malicious event record being missed and classified as non-malicious, the better for DCE with the model, data combination of Elastic Net Balanced having the highest FNR of 0.0164 while Random Forest Unbalanced having the highest FNR of 0.0551. Based on the data, model combination of Elastic Net Balanced have the lowest FNR of 0.0164, while Random Forest Unbalanced had the highest FNR of 0.0551.

5 Discussion

The best model, data combination in terms of accuracy, correctly classifying either of malicious or non-malicious events was of the Random Forest Balanced at 0.9993. Accuracy might be influenced by imbalance with the Unbalance data, model combination of Random Forest Unbalanced having slightly less but higher accuracy of 0.9987. The `Random Forest` model is at best suitable, in terms of accuracy and having associated lower chance of marking an event malicious when it is non-malicious (FPR), and marking a malicious event record being missed and classified as non-malicious (FNR). Based on variable importance, Table 4, `Assembled Payload Size` highly ranked on both data for Random forest, while for logistic elastic net, the `Server Response Packet Time` and `Connection State:Others` for balanced and unbalanced respectivey.

The DCE seeks to better keep malicious as malicious as compared to even have a single malicious within the non-malicious as it would be costly to their systems and networks. The false positive rate needs to be at a minimum, as such with the best model, data combination on FPR was of the Elastic Net Unbalanced at 0.

6 Conclusions

The earlier the detection of an event as malicious, the better for DCE as they would be able to put in place measure for risk analysis. However, it would be in the best interest for DCE to have fewer of non-malicious marked as malicious, but none of the malicious fall within the non-malicious, guided by the associated costs on the effects of attack on their systems and computer networks.

7 References

A. Ramezan, C., A. Warner, T., & E. Maxwell, A. (2019). Evaluation of sampling and cross-validation tuning strategies for Regional-Scale Machine Learning Classification. Remote Sensing, 11(2), 185. https://doi.org/10.3390/rs11020185

Fawagreh, K., Gaber, M. M., & Elyan, E. (2014). Random forests: from early developments to recent advancements. Systems Science & Control Engineering: An Open Access Journal, 2(1), 602-609.

F.Y, O., J.E.T, A., O, A., J. O, H., O, O., & J, A. (2017). Supervised machine learning algorithms: Classification and comparison. International Journal of Computer Trends and Technology, 48(3), 128–138. https://doi.org/10.14445/22312803/ijctt-v48p126

Hofstede, R., Jonker, M., Sperotto, A., & Pras, A. (2017). Flow-based web application brute-force attack and compromise detection. Journal of Network and Systems Management, 25(4), 735–758. https://doi.org/10.1007/s10922-017-9421-4

Lai, Y. P., & Hsia, P. L. (2007). Using the vulnerability information of computer systems to improve the network security. Computer Communications, 30(9), 2032-2047.

Liaw, A., & Wiener, M. (2002). Classification and regression by randomForest. R news, 2(3), 18-22.

Lipson, H. F. (2002). Tracking and tracing cyber-attacks: Technical challenges and global policy issues.

Yang, L., & Shami, A. (2020). On hyperparameter optimization of Machine Learning Algorithms: Theory and practice. Neurocomputing, 415, 295–316. https://doi.org/10.1016/j.neucom.2020.07.061

8 Appendix

8.1 Tables

Table 1: Hyper-parameter tuning

	mtry	num_tree
Random Forest Balanced	3.464102	500
Random Forest Unbalanced	12.000000	500

	alpha	lambda
Logistic Elastic Net Balanced	0.1	4.229243
Logistic Elastic Net Balanced1	0.0	0.010000

Table 2: Confusion Matrix

	Actual Values							
Predicted	Balanced		Unbalanced		Balanced		Unbalanced	
	Logistic Elastic Net		Logistic Elastic Net		Random Forest		Random Forest	
	NonMal	Mal	NonMal	Mal	NonMal	Mal	NonMal	Mal
NonMal	464392	45	469293	997	469148	206	468827	151
Mal	4903	2696	2	1744	147	2535	468	2590

Table 3: Perfomance Evaluation

	Logistic Elastic Net		Random Forest	
Metric	Balanced	Unbalanced	Balanced	Unbalanced
Accuracy	0.9895	0.9979	0.9993	0.9987
Sensitivity	0.9836	0.6363	0.9248	0.9449
PPV	0.3548	0.9989	0.9452	0.8470
FNR	0.0164	0.3637	0.0752	0.0551
FPR	0.0104	0.0000	0.0003	0.0010
F1	0.5215	0.7774	0.9349	0.8933

Table 5: Variable Description

variable	decription
Assembled Payload Size (continuous)	The total size of the inbound suspicious payload.
DYNRiskA Score (continuous)	An un-tested in-built risk score assigned by a new SIE
Response Size (continuous)	The total size of the reply data in the TCP conversati
Source Ping Time (ms) (continuous)	The 'ping' time to the IP address which triggered the
Operating System (Categorical)	A limited 'guess' as to the operating system that gene
Connection State (Categorical)	An indication of the TCP connection state at the tim
Connection Rate (continuous)	The number of connections per second by the inbound
Ingress Router (Binary)	DCE has two main network connections to the 'world
Server Response Packet Time (ms) (continuous)	An estimation of the time from when the payload was

Table 4: Variable Importance

	Random Forest		Logistic Elastic Net	
	Balanced	Unbalanced	Balanced	Unbalanced
'Assembled Payload Size'	100.0000	100.0000	1.3562	0.0146
'DYNRiskA Score'	13.7146	1.1848	0.0000	83.0023
'Response Size'	1.0735	0.2407	0.0000	0.0000
'Source Ping Time'	1.1170	0.0245	0.0000	0.0609
'Operating System'Others	0.0051	0.0000	0.0000	20.1555
'Operating System'Windows_All	0.0223	0.0024	0.0000	0.7350
'Connection State'Others	12.3687	0.0001	0.0000	100.0000
'Connection Rate'	5.9652	0.0589	0.0000	0.0449
'Ingress Router'syd-tls-04	0.0000	0.0004	0.0000	1.0883
'Server Response Packet Time'	72.9411	15.7459	100.0000	2.5491
'Packet Size'	1.5538	0.1912	0.0000	0.0814
'Packet TTL'	1.0510	0.0476	0.0000	0.3897
'Source IP Concurrent Connection'	4.3517	2.2339	0.0000	0.8626

8.2 Confusion Matrix Percentages

8.2.1 Logistice Elastic Net Balanced

```
##
##    Cell Contents
## |-------------------------|
## |                   Count |
## |          Column Percent |
## |-------------------------|
##
## Total Observations in Table:  472036
```

```
##
##                              | mydata.test$Class
## elastic_predictions_mydata_b |   NonMal  |      Mal  | Row Total |
## -----------------------------|-----------|-----------|-----------|
##                       NonMal |   464392  |       45  |   464437  |
##                              |       99% |        2% |           |
## -----------------------------|-----------|-----------|-----------|
##                          Mal |     4903  |     2696  |     7599  |
##                              |        1% |       98% |           |
## -----------------------------|-----------|-----------|-----------|
##                 Column Total |   469295  |     2741  |   472036  |
##                              |       99% |        1% |           |
## -----------------------------|-----------|-----------|-----------|
##
##
```

8.2.2 Logistice Elastic Net Unbalanced

```
##
##    Cell Contents
## |-------------------------|
## |                   Count |
## |          Column Percent |
## |-------------------------|
##
## Total Observations in Table:  472036
##
##                               | mydata.test$Class
## elastic_predictions_mydata_ub |   NonMal  |      Mal  | Row Total |
## ------------------------------|-----------|-----------|-----------|
##                        NonMal |   469293  |      997  |   470290  |
##                               |      100% |       36% |           |
## ------------------------------|-----------|-----------|-----------|
##                           Mal |        2  |     1744  |     1746  |
##                               |        0% |       64% |           |
## ------------------------------|-----------|-----------|-----------|
##                  Column Total |   469295  |     2741  |   472036  |
##                               |       99% |        1% |           |
## ------------------------------|-----------|-----------|-----------|
##
##
```

8.2.3 Random Forest Balanced

```
##
```

```
##    Cell Contents
## |-------------------------|
## |                   Count |
## |          Column Percent |
## |-------------------------|
##
## Total Observations in Table:  472036
##
##                         | mydata.test$Class
## rf_predictions_mydata_b |   NonMal  |      Mal  | Row Total |
## ------------------------|-----------|-----------|-----------|
##                  NonMal |   469148  |      206  |   469354  |
##                         |      100% |        8% |           |
## ------------------------|-----------|-----------|-----------|
##                     Mal |      147  |     2535  |     2682  |
##                         |        0% |       92% |           |
## ------------------------|-----------|-----------|-----------|
##            Column Total |   469295  |     2741  |   472036  |
##                         |       99% |        1% |           |
## ------------------------|-----------|-----------|-----------|
##
##
```

8.2.4 Random Forest Unbalanced

```
##
##    Cell Contents
## |-------------------------|
## |                   Count |
## |          Column Percent |
## |-------------------------|
##
## Total Observations in Table:  472036
##
##                          | mydata.test$Class
## rf_predictions_mydata_ub |   NonMal  |      Mal  | Row Total |
## -------------------------|-----------|-----------|-----------|
##                   NonMal |   468827  |      151  |   468978  |
##                          |      100% |        6% |           |
## -------------------------|-----------|-----------|-----------|
##                      Mal |      468  |     2590  |     3058  |
##                          |        0% |       94% |           |
## -------------------------|-----------|-----------|-----------|
##             Column Total |   469295  |     2741  |   472036  |
##                          |       99% |        1% |           |
## -------------------------|-----------|-----------|-----------|
##
##
```